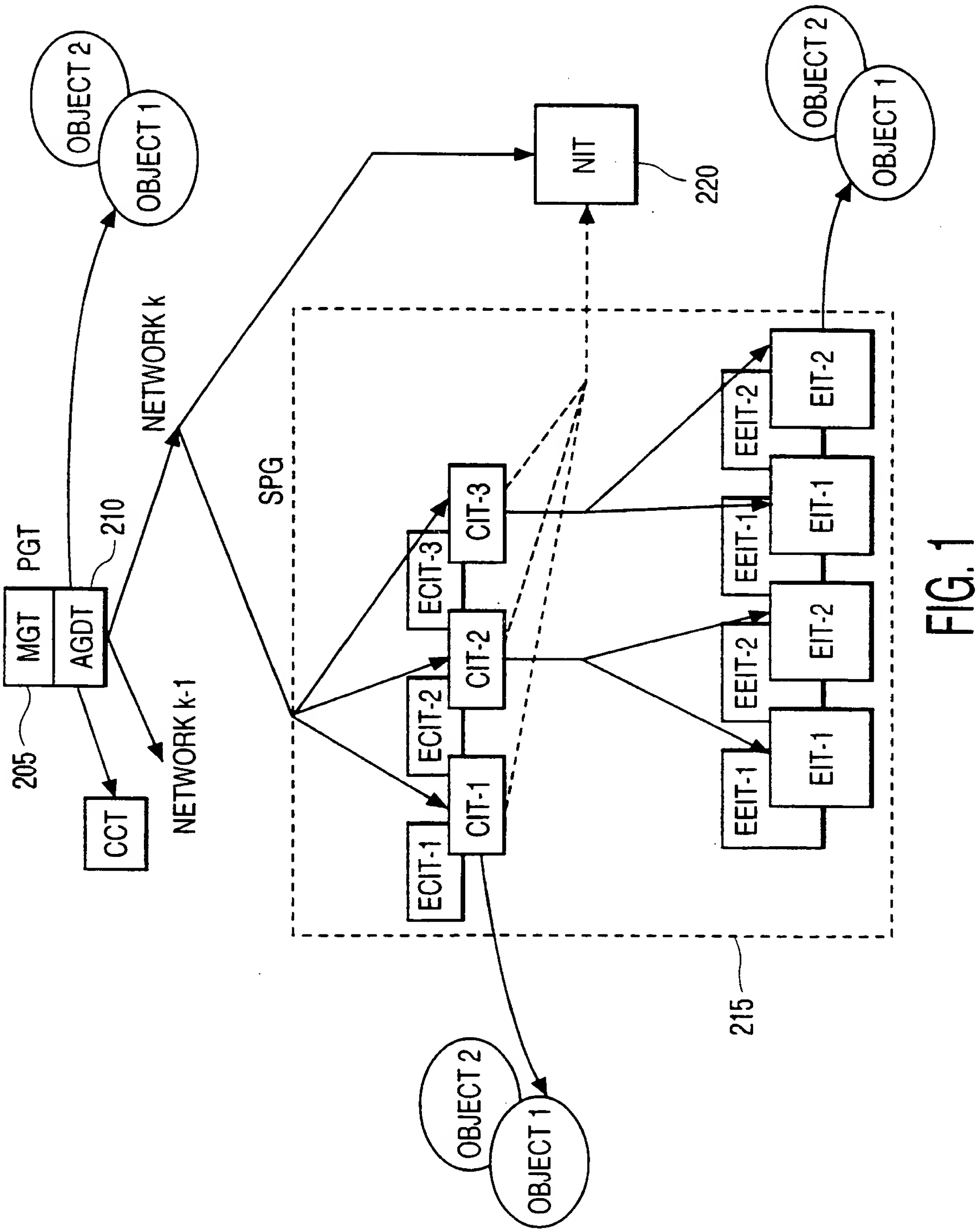
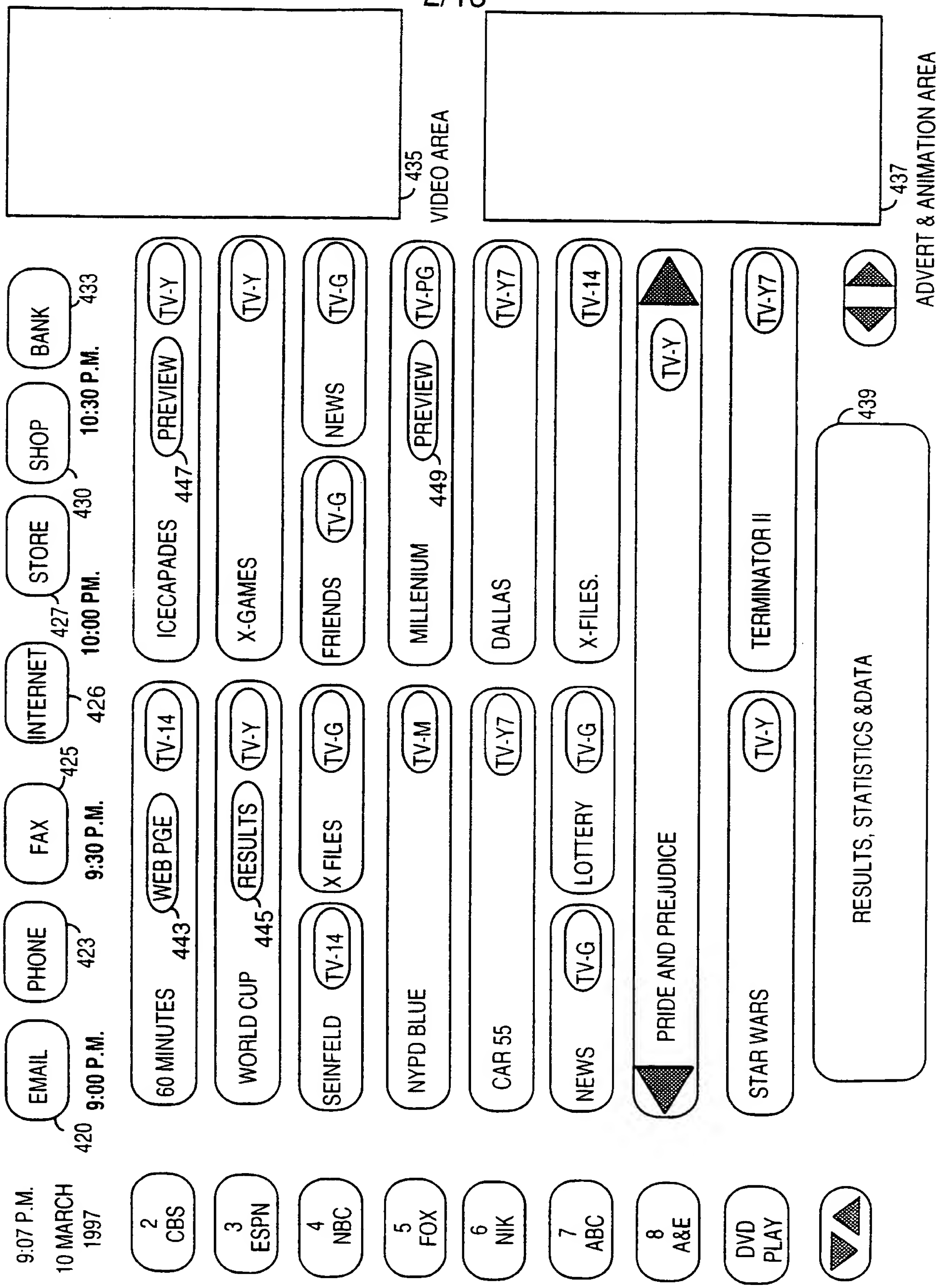

OBJECT 2
OBJECT 1
PGT
MGT
AGDT
205
210
CCT
NETWORK k-1
NETWORK k
NIT
220
SPG
ECIT-1
CIT-1
ECIT-2
CIT-2
ECIT-3
CIT-3
EEIT-1
EIT-1
EEIT-2
EIT-2
EEIT-1
EIT-1
EEIT-2
EIT-2
OBJECT 2
OBJECT 1
OBJECT 2
OBJECT 1
215

FIG. 1

FIGURE 2


9:07 P.M.
10 MARCH
1997
EMAIL
420
PHONE
423
FAX
425
INTERNET
426
STORE
427
SHOP
430
BANK
433
9:00 P.M.
9:30 P.M.
10:00 PM.
10:30 P.M.
435
VIDEO AREA
2
CBS
60 MINUTES
443
WEB PGE
TV-14
ICECAPADES
447
PREVIEW
TV-Y
3
ESPN
WORLD CUP
445
RESULTS
TV-Y
X-GAMES
TV-Y
4
NBC
SEINFELD
TV-14
X FILES
TV-G
FRIENDS
TV-G
NEWS
TV-G
5
FOX
NYPD BLUE
TV-M
MILLENIUM
449
PREVIEW
TV-PG
6
NIK
CAR 55
TV-Y7
DALLAS
TV-Y7
7
ABC
NEWS
TV-G
LOTTERY
TV-G
X-FILES.
TV-14
8
A&E
PRIDE AND PREJUDICE
TV-Y
DVD
PLAY
STAR WARS
TV-Y
TERMINATOR II
TV-Y7
437
ADVERT & ANIMATION AREA
RESULTS, STATISTICS &DATA
439

	SYNTAX	BITS	FORMAT
	MGT_message () {		
	reserved	2	'11'
	life_time	22	uimsbf
	current_time	40	uimsbf
300	Num_bytes_AGDT	16	uimsbf
	}		

FIG. 3

	SYNTAX	BITS	FORMAT
	AGDT_message () {		
	reserved	3	'111'
	CCT_version	5	uimsbf
	reserved	4	'1111'
	EPG_descriptors_length	12	uimsbf
	for (i=0;i<N;i++) {		
405	descriptor ()	var	
	}		
	num_bytes_CCT	16	uimsbf
	number_of_networks	8	uimsbf
	for (i = 0 ; i < number_of_networks; i++){		
	reserved	3	'111'
	NIT_version	5	uimsbf
	num_bytes_NIT[i]	16	uimsbf
	reserved	4	'1111'
	network_descriptors_length	12	uimsbf
	for (i=0;i<N;i++){		
410	descriptor ()		
	}		
415	program_guide_map ()	var	
	}		
	}		

FIG. 4

```
      SYNTAX                                                          BITS      FORMAT

      program_guide_map () {
          number_channel_groupings                                    4         uimsbf
          SPG_map_descriptors_length                                  12        uimsbf
          for (i=0;I<N;i++) {
505           descriptor ()                                           var
          }
          for (i = 0;I<number_channel_groupings+1;i++) {
              reserved                                                4         '1111'
              start_channel(i)                                        12        uimsbf
          }
          number_guides                                               8         uimsbf
          reserved                                                    4         '1111'
          program_guide_map_size                                      12        uimsbf
510       for (i = 0; i< number_guides+1;i++) SPG_map(i) {
              next                                                    8         uimsbf
              previous                                                8         uimsbf
              left_column_time                                        40        bslbf
              width_in_minutes                                        16        uimsbf
              reserved                                                4         '1111'
              SPG_descriptors_length                                  12        uimsbf
              for (i=0;i<N;i++) {
515               descriptor ()                                       var
              }
              Nbytes_list_SPG (i) {
520           for (j = 0;j< number_channel_groupings+1;j++)
                  {
                      reserved                                        4         '1111'
                      group[j]_descriptors_length                     12        uimsbf
                      for (I=0;I<N;I++) {
525                       descriptor ()                               var
                      }
                      Num_bytes_SPG[i]_CIT[j]                         16        uimsbf
                      Num_bytes_SPG[i]_ECIT[j]                        16        uimsbf
                      Num_bytes_SPG[i]_EIT[j]                         16        uimsbf
                      Num_bytes_SPG[i]_EEIT[j]                        16        uimsbf
                  }
              }
              SPG_name_length                                         8         uimsbf
              for(i=0;i< SPG_name_length;i++)
                  SPG_name(i)                                         8         ISO-639
          }
      }
```

FIG. 5

	SYNTAX	BITS	FORMAT
	multimedia object descriptor() {		
	descriptor_tag	8	0x5F
	descriptor_length	8	uimsbf
605	object_type	8	uimsbf
	if (object _type = 0xFF) {		
	extended_object_type	16	uimsbf
	}		
610	address_descriptor		
	object_format	8	uimsbf
	object_version_number	7	uimsbf
	display_mode	1	0/1
	object_start_time	40	uimsbf
	object_duration_format	2	uimsbf
	object_duration	14	uimsbf
	object_frame_size	32	uimsbf
	}		

FIG. 6

ELEMENT	DEFINITION
descriptor_tag	SET TO 0x5F TO IDENTIFY THE DESCRIPTOR AS AN OBJECT DESCRIPTOR.
descriptor_length	DESCRIPTOR LENGTH IN BYTES FOLLOWING THIS FIELD.
object_type and extended_object_type	SPECIFIES OBJECT TYPE.
address_descriptor	OBJECT ADDRESS.
object_format	OBJECT FORMAT.
object_version_number	SPECIFIES THE CURRENT VERSION OF THE OBJECT. AN APPLICATION, FOR EXAMPLE CAN USE THIS FIELD TO DETERMINE WHETHER IT SHOULD RELOAD THE OBJECT THAT IS ALREADY PRESENT IN THE BOX.
display mode	THIS FIELD CAN EITHER BE "ON-DEMAND"(0) OR "IMMEDIATE"(1). WHEN AN "IMMEDIATE" OBJECT BECOMES "ALIVE" AS DETERMINED BY THE Object_start_time, WE SHOULD IMMEDIATELY NOTIFY THE USER ABOUT THE AVAILABILITY. E.g.: AN OBJECT ASSOCIATED WITH A COMMERCIAL THAT IS BEING AIRED. THE AVAILABILITY OF AN "ON DEMAND" OBJECT IS NOTIFIED TO THE USER ONLY WHEN THE USER WANTS TO SEE THE AVAILABLE OBJECTS LIST.
object_start_time	SPECIFIES THE TIME AT WHICH THE OBJECT BECOMES "ALIVE". THE OBJECT IS AVAILABLE FOR THE USER STARTING FROM THIS TIME.
object_duration_format	IF THE VALUE IS 1/2/3/4 THEN THE object_duration IS IN SECONDS, MINUTES, HOURS, OR DAYS RESPECTIVELY.
object_duration	SPECIFIES THE TIME AT WHICH THE OBJECT EXPIRES.
object_frame_size	OBJECT FRAME SIZE IN BYTES. Object_frame CONSISTS OF THE object_header AND THE ACTUAL OBJECT.

FIG. 7

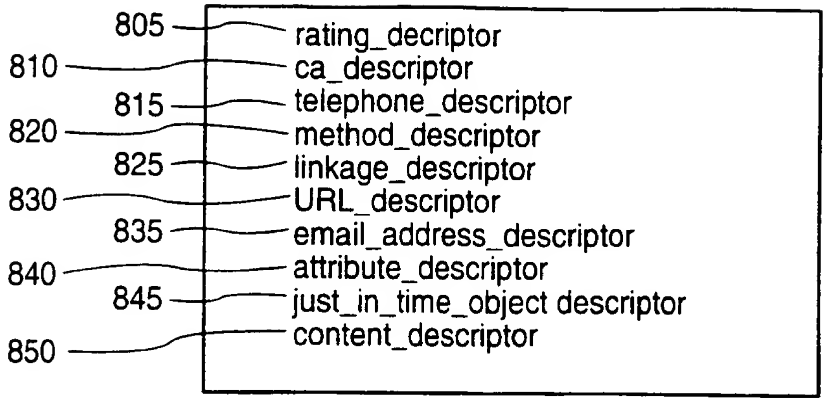


FIG. 8

ELEMENT	DEFINITION
rating_descriptor	THE rating_descriptor SPECIFIES THE PARENTAL RATING FOR THE OBJECT.
ca_descriptor	THE ca_descriptor SPECIFIES THE CONDITIONAL ACCESS SYSTEM FOR THE OBJECT.
telephone_descriptor	THE telephone_descriptor SPECIFIES THE TELEPHONE NUMBER AND RELATED INFORMATION ASSOCIATED WITH THE OBJECT.
method_descriptor	THE method_descriptors ASSOCIATED WITH AN OBJECT DESCRIBE THE METHODS AND THE EVENTS THAT WILL TRIGGER THEM.
linkage_descriptor	THE linkage_descriptor LINKS OTHER DESCRIPTORS TO THE CURRENT OBJECT DESCRIPTOR.
attribute_descriptor	THE attribute_descriptor SHALL BE USED TO SPECIFY THE SPECIAL ATTRIBUTES OF THE CURRENT OBJECT.
just_in_time_object descriptor	THIS DESCRIPTOR IS USED TO INDICATE THE ADDRESS OF THE MODs AND OBJECTS THAT ARE NOT KNOWN IN ADVANCE.
content_descriptor	THIS DESCRIPTOR IS USED TO SPECIFY THE OBJECTS PROFILE VALUES FOR TARGETTED COMMERCIALS.

FIG. 9

	SYNTAX	BITS	FORMAT
	remote_http_object_address_descriptor() {		
	descriptor_tag	8	uimsbf
	descriptor_length	8	uimsbf
	URL_length	8	uimsbf
	for (i = 0;i<URL_length;i++) {		
905	URL(i)	8	ISO-639
	}		
	}		

FIG. 10

	SYNTAX	BITS	FORMAT
	DSM-CC_object_address_descriptor() {		
	descriptor_tag	8	uimsbf
	descriptor_length	8	uimsbf
910	DSM-CC_association_tag	16	uimsbf
	}		

FIG. 11

	SYNTAX	BITS	FORMAT
	MPEG_PSI_PS_address_descriptor() {		
	descriptor_tag	8	uimsbf
	descriptor_length	8	uimsbf
	default_primary_location_bit	1	0/1
	if (default_primary_location_bit == 0) {		
915	network_id	8	uimsbf
920	transport_channel_id	8	uimsbf
	}		
	default_secondary_location_bit	1	0/1
	if (default_secondary_location_bit == 0) {		
925	PID	13	uimsbf
930	table_id	8	uimsbf
	table_id_extension	16	uimsbf
	}		
	}		

FIG. 12

```
     SYNTAX                                                   BITS     FORMAT

     descriptor_tag                                            8       uimsbf
     descriptor_length                                         8       uimsbf
950  number_elements                                           8       uimsbf
     for (i=0;i<number_elements;i++) {
         reserved                                              3       '111'
         size_flag                                             1       uimsbf
955      element_identifier                                    12      uimsbf
         if (transport == broadcast) {
960          transport_channel_ID                              8       uimsbf
             reserved                                          3       '111'
965          PID                                               13      uimsbf
         }
         else if (transport == file based) {
             file_name_length                                  8       uimsbf
             for (i=0;i<address_length;i++)
970              file_char                                     8       ISO-639
         }
         if (size_flag == 1) {
             element_size                                      32      uimsbf
     }
```

FIG. 13

element_identifier	description
0x000	user private
0x001	Private Information Parcel (PIP)
0x002	Extended Text Table (ETT)
0x003	Network Information Table (NIT)
0x004	Special Program Guide (SPG)
0x005	Channel Information Table (CIT)
0x006	Extented Channel Information Table (ECIT)
0x007	Event Information Table (EIT)
0x008	Extended Event Information Table (EEIT)

FIG. 14

```
     SYNTAX                                          BITS      FORMAT

     location_descriptor () {
980      descriptor_tag                               8        uimsbf
         descriptor_length                            8        uimsbf
         number_PIDs                                  8        uimsbf
         reserved                                     7        '1111111'
         implicit_flag                                1        bslbf
985      if (implicit_flag == 0x00){
987          for (i=1;i<number_PIDs;I++){
                 reserved                             3        '111'
990              PID[i]                               13       uimsbf
                 SType[i]                             8        uimsbf
             }
         } else {
             reserved                                 3        '111'
993          base_PID                                 13       uimsbf
         }
     }
```

FIG. 15

```
     SYNTAX                                                  BITS      FORMAT

     location_descriptor () {
350      descriptor_tag                                       8        uimsbf
         descriptor_length                                    8        uimsbf
         number_SCIDs                                         8        uimsbf
         reserved                                             6        '111111'
         Z_bit                                                1        bslbf
         implicit_flag                                        1        bslbf
353      if (implicit_flag == 0x00){
             for (i=1;i<number_SCIDs;i++){
                 if (Z_bit==0)
355                  SCID[i]                                  8        uimsbf
                 else{
                     reserved                                 4        '1111'
357                  SCID[i]                                  12       uimsbf
                 }
                 SType[i]                                     8        uimsbf
             }
         } else {
             if (Z_bit==0)
360              base_SCID                                    8        uimsbf
             else{
                 reserved                                     4        '1111'
363              base_SCID                                    12       uimsbf
             }
         }
     }
```

FIG. 16

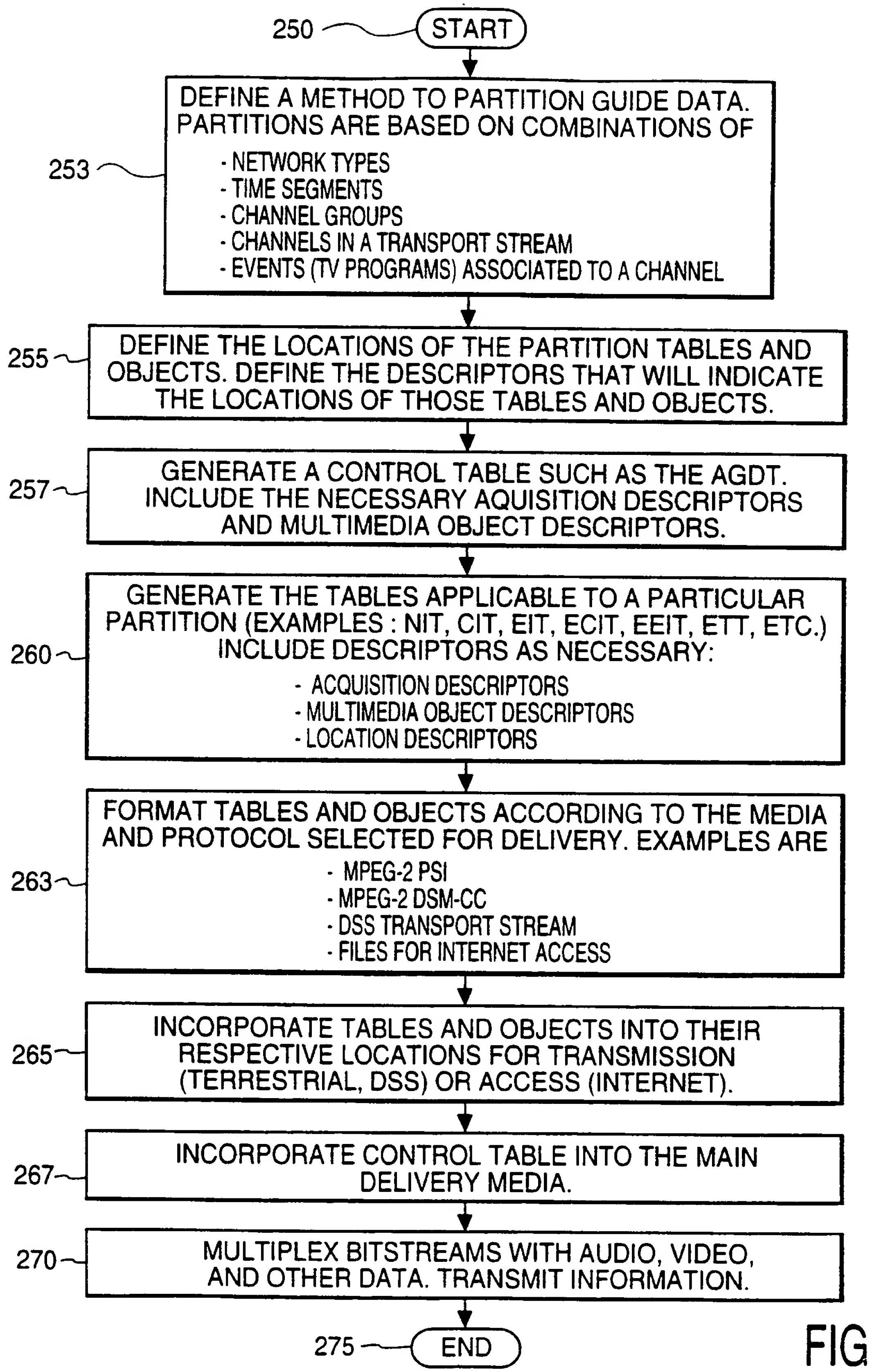

250
START
253
DEFINE A METHOD TO PARTITION GUIDE DATA. PARTITIONS ARE BASED ON COMBINATIONS OF
- NETWORK TYPES
- TIME SEGMENTS
- CHANNEL GROUPS
- CHANNELS IN A TRANSPORT STREAM
- EVENTS (TV PROGRAMS) ASSOCIATED TO A CHANNEL
255
DEFINE THE LOCATIONS OF THE PARTITION TABLES AND OBJECTS. DEFINE THE DESCRIPTORS THAT WILL INDICATE THE LOCATIONS OF THOSE TABLES AND OBJECTS.
257
GENERATE A CONTROL TABLE SUCH AS THE AGDT. INCLUDE THE NECESSARY AQUISITION DESCRIPTORS AND MULTIMEDIA OBJECT DESCRIPTORS.
260
GENERATE THE TABLES APPLICABLE TO A PARTICULAR PARTITION (EXAMPLES : NIT, CIT, EIT, ECIT, EEIT, ETT, ETC.) INCLUDE DESCRIPTORS AS NECESSARY:
- ACQUISITION DESCRIPTORS
- MULTIMEDIA OBJECT DESCRIPTORS
- LOCATION DESCRIPTORS
263
FORMAT TABLES AND OBJECTS ACCORDING TO THE MEDIA AND PROTOCOL SELECTED FOR DELIVERY. EXAMPLES ARE
- MPEG-2 PSI
- MPEG-2 DSM-CC
- DSS TRANSPORT STREAM
- FILES FOR INTERNET ACCESS
265
INCORPORATE TABLES AND OBJECTS INTO THEIR RESPECTIVE LOCATIONS FOR TRANSMISSION (TERRESTRIAL, DSS) OR ACCESS (INTERNET).
267
INCORPORATE CONTROL TABLE INTO THE MAIN DELIVERY MEDIA.
270
MULTIPLEX BITSTREAMS WITH AUDIO, VIDEO, AND OTHER DATA. TRANSMIT INFORMATION.
275
END

FIG. 17

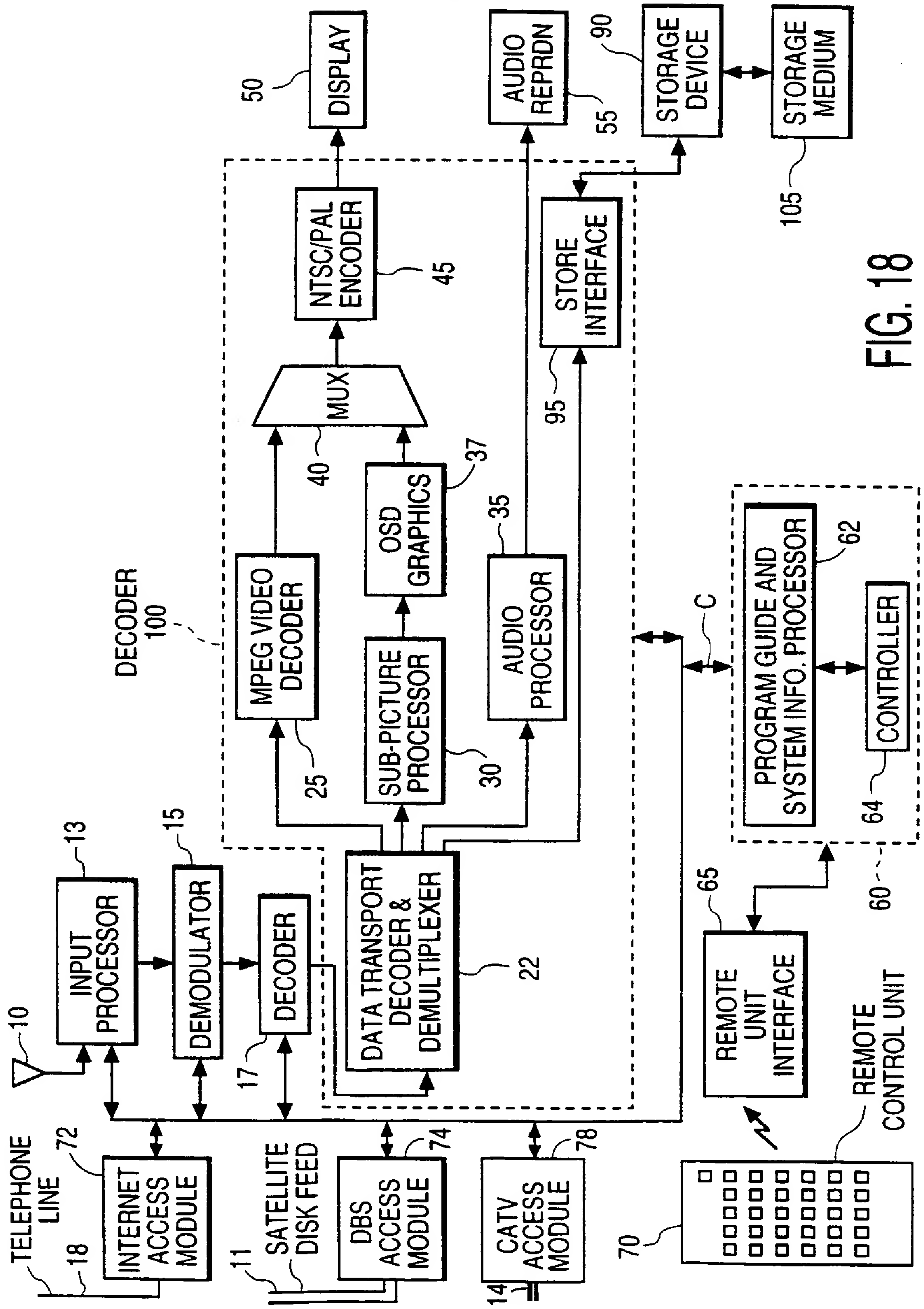

TELEPHONE LINE
18
72
INTERNET ACCESS MODULE
11
SATELLITE DISK FEED
DBS ACCESS MODULE
74
14
CATV ACCESS MODULE
78
10
INPUT PROCESSOR
13
DEMODULATOR
15
DECODER
17
DATA TRANSPORT DECODER & DEMULTIPLEXER
22
DECODER 100
MPEG VIDEO DECODER
25
SUB-PICTURE PROCESSOR
30
OSD GRAPHICS
37
AUDIO PROCESSOR
35
MUX
40
NTSC/PAL ENCODER
45
DISPLAY
50
AUDIO REPRDN
55
STORE INTERFACE
95
STORAGE DEVICE
90
STORAGE MEDIUM
105
C
PROGRAM GUIDE AND SYSTEM INFO. PROCESSOR
62
CONTROLLER
64
60
REMOTE UNIT INTERFACE
65
REMOTE CONTROL UNIT
70

FIG. 18